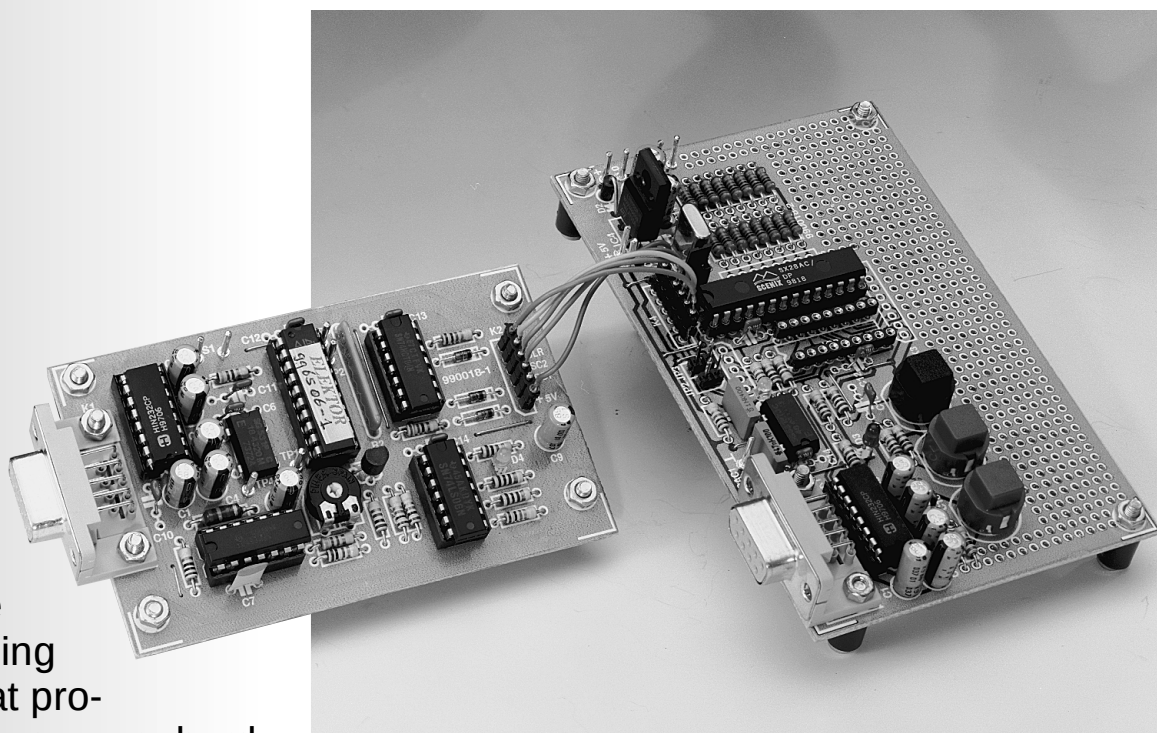


SX-microcontroller evaluation system (3)

part 3: the SX prototyping board

This month we present the SX prototyping board that provides the necessary hardware and a prototyping area for your own experiments. Add the PICKALOCK programming adaptor and you are able to use a PC to load programs into the SX microcontroller. The prototyping board is suitable for 18 as well as 28-pin versions of the Scenix SX microcontroller.



The prototyping board was designed to help you learn about practical SX programming and system development. The circuit diagram is shown in **Figure 1**. Originally, separate boards were designed for the two versions of the SX controller (SX18 and SX28). However, as the design was gradually finished, it was decided to make the system even more versatile by combining the two boards.

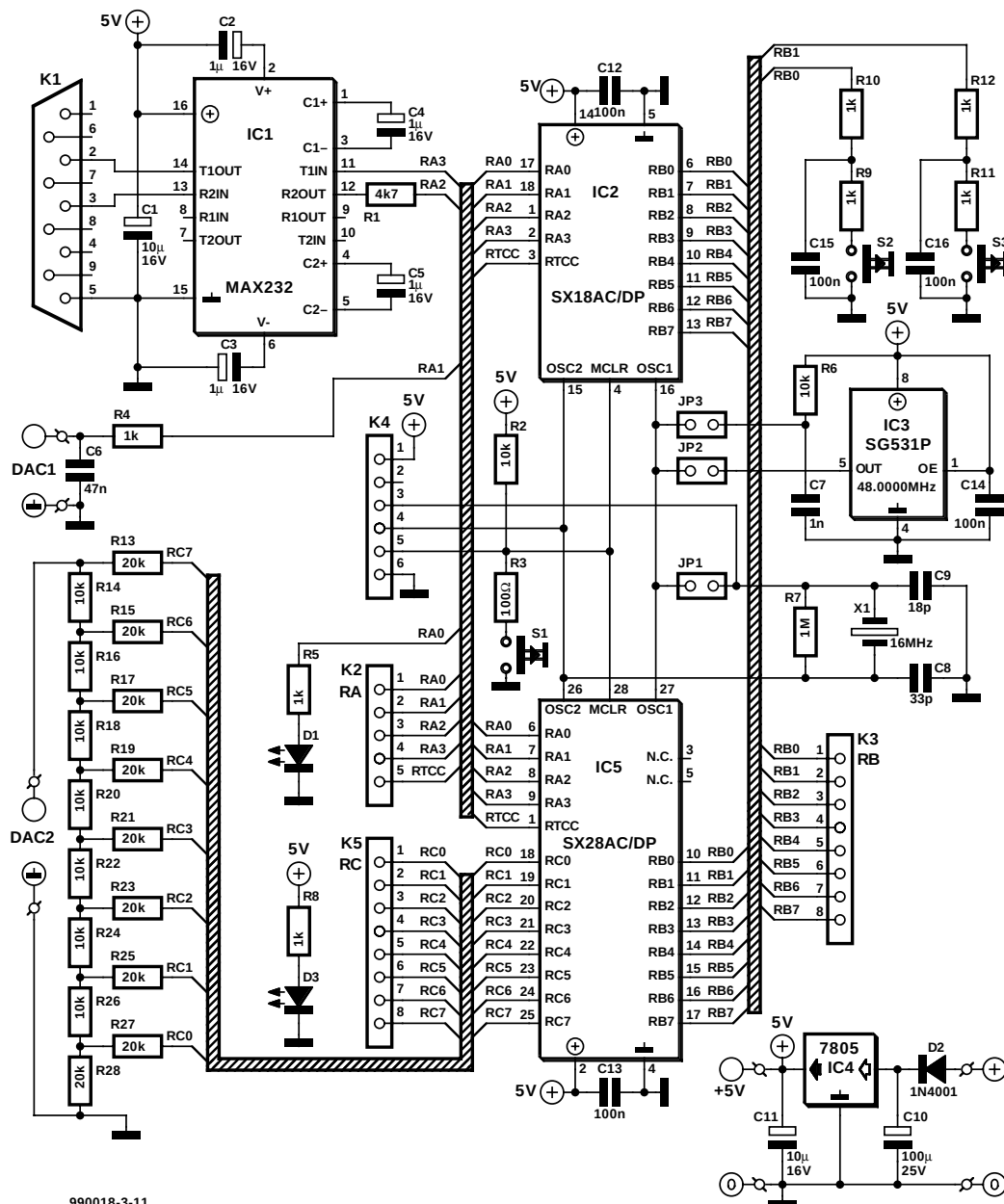
OSCILLATOR HARDWARE

The prototyping board offers three options for the oscillator configuration. Normally, jumper **JP1** is fitted, enabling the internal oscillator and its satellite parts X1, R7, C8 and C9 to act as a 16-MHz clock generator. If, however, jumper **JP2** is fitted, the SX chip is clocked at 48 MHz by integrated clock oscillator IC3. These two frequencies were selected for the demonstration programs. For your own experiments, it is, of course, possible to use other quartz crystals or external oscillator modules. If jumper **JP3** is fitted, the R-C network R6-C7 is connected to the OSC1 oscillator pin. In this configuration, the RC oscillator option of the SX microcontroller is employed.

ISP ADAPTOR

With jumper **JP1** installed, the SX micro

Design by M. Ohsmann



990018-3-11

Figure 1. Circuit diagram of the SX prototyping board.

may be connected to the SX-PICK-ALOCK programmer via the ISP connector, K4. In this configuration it is possible to re-program the SX chip 'in-circuit'. The connecting cable should be not be too long to ensure that the 16-MHz oscillator works reliably with the programming adaptor connected- up.

SIMPLE I/O

Port RA provides a simple input/output interface on the prototyping board. An LED is connected to port line RA.0, and an RC network to RA.1. The latter enables analogue voltages to be generated through pulsewidth modulation (PWM). Port lines RA.2 and RA.3 are linked to the MAX232 RS232 interface circuit (IC1), enabling your programs to communicate with a PC (or a dumb terminal) via a serial cable connected to K1. A simple D-A converter is connected to port RC of the 28-pin version

of the SX micro. The DAC enables fast analogue signals to be generated under software control. Push-buttons S2 and S3 may be used as rudimentary input devices. They are connected to port RB to enable you to experiment with the possibilities of 'wake up on interrupt'. Because all processor I/O pins are accessible via pinheaders, they are easily connected to extension circuits built on the prototyping area of the board. A reset switch is also available: S1.

The board is powered by a 9-volt supply. The supply voltage connected to the prototyping board is available at a pin on K4, enabling a programming adaptor to be powered also.

CONSTRUCTION

The prototyping board is a single-sided type whose copper track layout and component mounting plan are given

in **Figure 2**. This board is available ready-made through the Publishers' Readers Services as well as through kit suppliers supporting Elektor projects. Although populating the board is not particularly difficult, the component placement and soldering should be carried out with great care and precision. Also, be sure to fit all wire links as indicated on the component overlay. Note that K1 is a 9-way sub-D **socket** (i.e., female connector). We recommend using IC sockets with turned pins for the SX micro-controllers (positions IC2 and IC5) because these will typically be fitted and removed quite a few times. The two SX micros **should never be on the board at the same time!**

Once the hardware is ready for use (see **Figure 3**), you may start thinking about the system software. However, before 'assembling' your first program

2

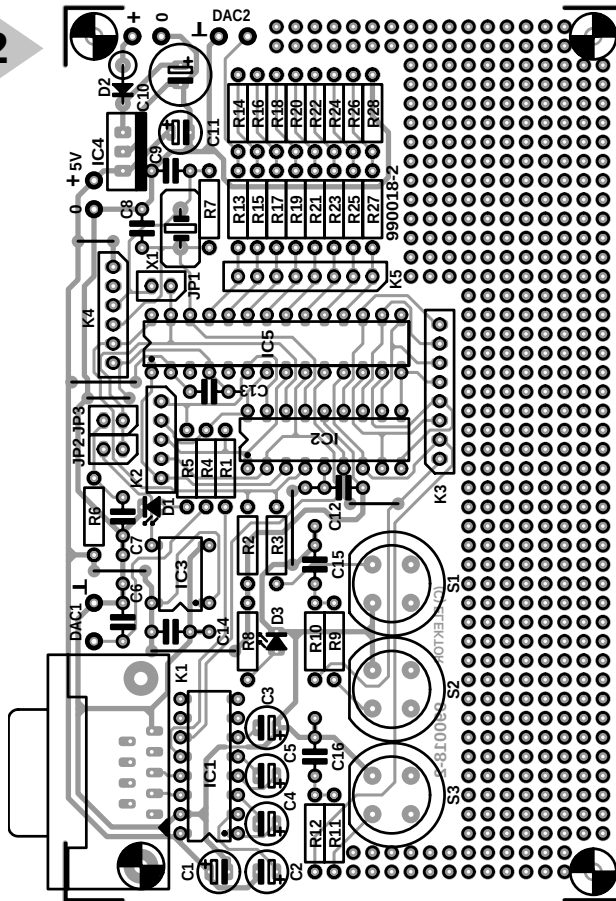


Figure 2. PCB copper track layout and component overlay (board available ready-made).

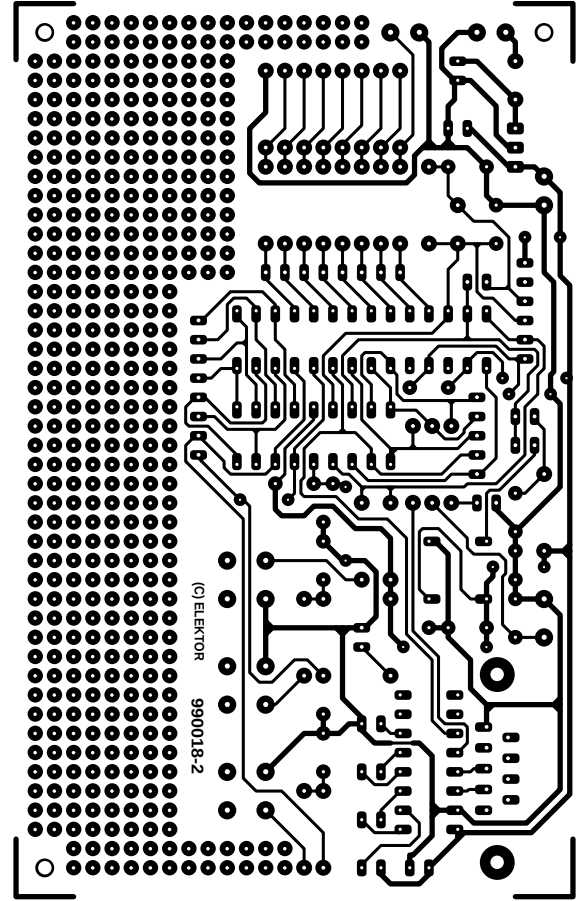
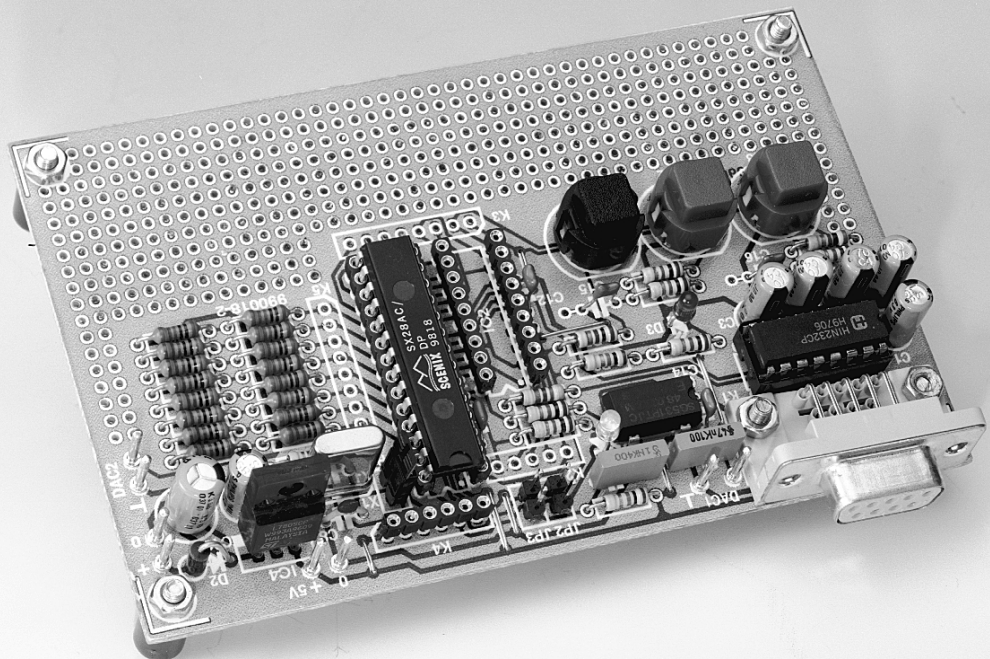


Figure 3. Ready populated prototype of the prototyping board.

3



COMPONENTS LIST

Resistors:

R1 = 4k Ω
 R2,R6 = 10k Ω
 R3 = 100 Ω
 R4,R5,R8-R12 = 1k Ω
 R7 = 1M Ω
 R14,R16,R18,R20,R22,R24,R26 = 10k Ω 1%
 R13,R15,R17,R19,R21,R23,R25,R27,R28 = 20k Ω 1%

Capacitors:

C1,C11 = 10 μ F 16V radial
 C2-C5 = 1 μ F 16V radial
 C6 = 47nF MKT
 C7 = 1nF MKT
 C8 = 33pF ceramic
 C9 = 18pF ceramic
 C10 = 100 μ F 25V radial
 C12-C16 = 100nF ceramic

Semiconductors:

D1,D3 = LED, high efficiency
 D2 = 1N4001
 IC1 = MAX232
 IC2 = SX18AC/DP*
 IC3 = SG531P 48.0000MHz (Epson) (Eurodis, Bolton)
 IC4 = 7805
 IC5 = SX28AC/DP*

Miscellaneous:

JP1,JP2,JP3 = 2-way pinheader w. jumper
 K1 = 9-way sub-D socket (female), angled pins, PCB mount
 K2 = 5-way SIL pinheader
 S1,S2,S3 = pushbutton D6R-RD (ITC) (Eurodis, Bolton)
 K3,K5 = 9-way SIL pinheader
 X1 = 16MHz quartz crystal
 K4 = 6-way SIL pinheader
 1 off 18-way IC socket with turned pins (for IC2)
 1 off 28-way IC socket with turned pins, 0.3-in pin distance (for IC5)
 PCB, order code 990018-2 (see Readers Services page in March 1999 issue)

on the PC, and actually program the first SX micro, it is useful to discuss the processor and the control software in some more detail.

SXASM ASSEMBLER AND SX INSTRUCTION SET

The SXASM assembler was developed to enable you to write your own programs for the SX micro. It is supplied on a 3.5-inch disk supplied for this project through our Readers Services. Because the disk also contains the assembler source code file, it should be possible to write your own extensions for the assembler. Admittedly, the assembler is not as powerful as the one supplied by Parallax, but that need not be a problem for everyday use. For example, SXASM does not support local macros or labels. However, the

standard macro instructions of the SX micro are available so that problems will rarely occur. A description of the assembler's features may also be found on the project diskette (see SXASM.DOC).

The instruction set is largely similar to that of the Microchip PIC microcontroller. Like the PIC, the SX controller requires you to take a hard look at the system of 'memory banks' in the RAM range, and the 'page' structure with jumps and subroutines. Because the SX is a RISC processor, many operations require the use of two instructions. Although similar operations use just one instruction on an 8051, the much higher clock rate of the SX ensures a very high processing speed.

YOUR FIRST PROGRAM

An example of an SX program as created with the aid of the SXASM assembler is shown in **Figure 4**. As you can see, the approach is traditional. Lines consist of an (optional) label, followed by an opcode, an operand and, (optional) comment. Opcodes and operands are case-insen-

sitive, i.e., they may be written in upper or lower case. Our very first program does nothing sensational except perhaps flash LED D1 on the prototyping board! The first program is always the most difficult because you will typically encounter lots of pitfalls. To keep 'clerical' errors like typos and syntax errors to a minimum, the diskette contains a number of tested example programs.

STEP BY STEP

Figure 5 illustrates the connections between the modules that make up the SX evaluation system. Start by reading the sequence described below — practical points will be discussed further on.

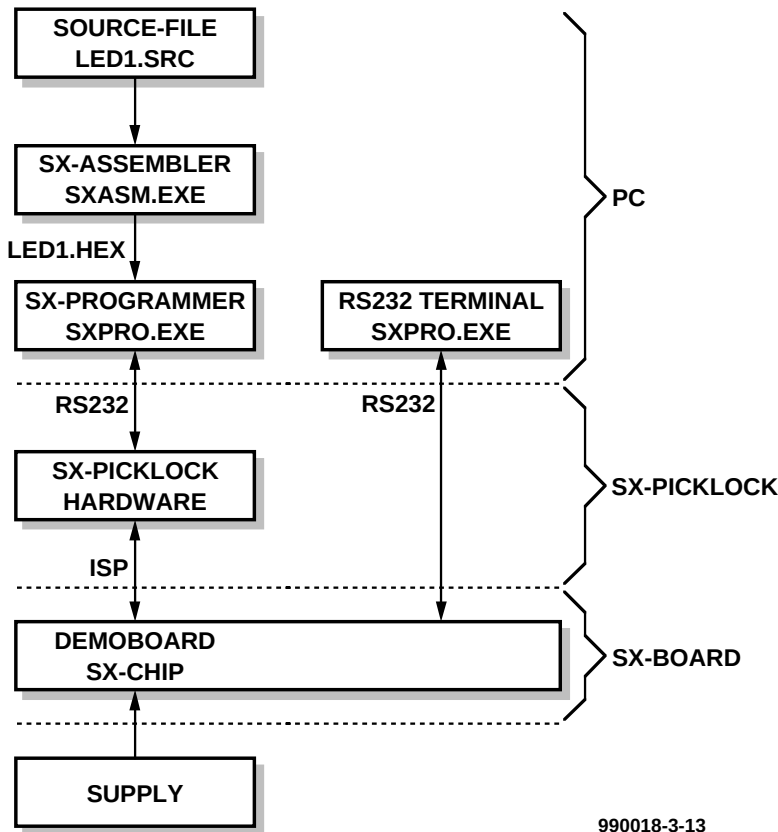
The source code text of the program (LED1.SRC) is assembled using the SX assembler (SXASM). This is done by typing

```
SXASM LED1 <return>
```

Figure 4. Try this program to get the feel: a LED flasher for the SX micro.

4

```
***** LISTING of SXASM (LED1) *****
LVL LINE LOC OBJ SOURCE
(0) 1 0000 device turbo,pins28
(0) 2 0000 divider equ 200 ; repetition of outer loop
(0) 3 0000 ra equ 5 ; define rort A
(0) 4 0000
(0) 5 0000 org 8 ; start of variable space in RAM
(0) 6 0008 cnt0 ds 1 ; define 3 counters as bytes
(0) 7 0009 cnt1 ds 1
(0) 8 000A cnt2 ds 1
(0) 9 000B
(0) 10 000B org 0 ; program memory starts at 0
(0) 11 0000 reset start ; reset vector jumps to start
(0) 12 0000
(0) 13 0000 C00 005 start mov !ra,#0; set port A to outputs
(0) 14 0002 907 loop call time ; call delay subroutine
(0) 15 0003 505 setb ra.0 ; turn on LED
(0) 16 0004 907 call time ; call delay again
(0) 17 0005 405 clrb ra.0 ; turn off LED
(0) 18 0006 A02 jmp loop ; start all over
(0) 19 0007
(0) 20 0007 CC8 02A time mov cnt2,#divider ; how often outer loop
(0) 21 0009 C1E 029 lp3 mov cnt1,#30 ; next level 30 times
(0) 22 000B 068 lp2 clr cnt0 ; inner level 256 times
(0) 23 000C 2E8 A0C lp1 djnz cnt0,lp1 ; inner loop
(0) 24 000E 2E9 A0B djnz cnt1,lp2 ; next level loop
(0) 25 0010 2EA A09 djnz cnt2,lp3 ; outer loop:count=divider
(0) 26 0012 00C ret
(0) 27 0013 end
FUSE = 07FA
FUSEX = 0F7E
RESET = 0000
***** SYMBOLTABLE (11 symbols) *****
divider :00C8 ra :0005 cnt0 :0008 cnt1 :0009
cnt2 :000A start :0000 loop :0002 time :0007
lp3 :0009 lp2 :000B lp1 :000C
(27 lines, 0 errors, 19 bytes)
```



990018-3-13

Figure 5. Block diagram of the modules that make up the SX Evaluation System.

This creates the hexadecimal file 'LED1.HEX' and the list file 'LED1.LST'. Next, activate the PC part, SXPRO.EXE, of your programming adaptor by typing

SXPRO LED1 <return>

This causes the SXPRO program to load the indicated HEX file and start communicating with the SX PICK-ALOCK programming adaptor via the RS232 link on the PC. The SX PICK-ALOCK hardware is connected to the prototyping board by way of the ISP (in-system programming) interface implemented on connector K1. Having connected all this hardware, you may switch on or plug in the 9-V mains adaptor. The prototyping board is then powered, and the SX microcontroller will start to operate. At the same time, however, the SX PICKALOCK is activated because it is also powered (via K4). At this point, SXPRO is allowed to send the SX program to the prototyping board by way of the ISP. This only requires typing

P <return>

in SXPRO (P for Program). Doing so

causes the file 'LED1.HEX' to be copied into the SX microcontroller. This program is executed after a reset, i.e., after you press S1.

So much for a general overview of the programming sequence, the next thing to do actually performing these steps. However, we should first take a look at the listing as shown in Figure 4.

OPTIONS AND INITIALISATION

The Scenix SX microcontroller has a number of options that have to be included when programming the FUSE and FUSEX registers. The configuration of these bits is implemented by the 'device' statement in line 1 of the program. In this example, the SX chip is switched to 'turbo' mode, which causes each instruction to be executed with a minimum of clock cycles. We also 'declare' that the 28-pin version is used. As to the oscillator type selection, no specification is required because a fast quartz crystal is employed. The program is stored on disk under the name LED1.SRC. It is first assembled by typing

SXASM LED1 <return>

Only then are you allowed to connect the PICKALOCK programming adaptor to the PC and the prototyping board, and apply power to the board. Next, launch the SXPRO program on the PC by typing

SXPRO LED1 <return>

The option 'P' in this program then allows you to actually load the SX chip with the designated program (attention: jumper JP1 has to be installed). Next, press the reset pushbutton on the board, S1, whereupon the LED should start to flash! If this works, the hardware may safely be assumed to work as specified, and you can start programming! Just in case the LED does not flash, inspect the hardware to locate any faults.

MACRO INSTRUCTIONS AND CLOCK CYCLES

Each simple instruction the SX micro is capable of executing has the size of a 'word' (12 bits). However, a close look at the listing in Figure 4 reveals that some instructions generate two words (for example, the one on line 13). These are macro instructions. Introduced by Parallax, macro instructions provide a kind of shorthand notation similar to that used with PIC processors. Be careful, because many macro commands 'invisibly' modify the W register or the flags. Some of these were even modified by the CARRYX option. If you want to know all the details, study the SX documentation in great detail, there is no other way! In addition to the documentation supplied by Parallax and Scenix, the project diskette (order code 996007-1) also provides useful information. It should also be noted that the macro instructions require more than one cycle. The same goes for jumps and subroutines, however all this should be familiar from other RISC processors. Consequently, a speed comparison between the SX processor and other processors is not that simple. In the example program discussed above, three 8-bit variables are declared (set up) in RAM in lines 5 to 9, next, port RA (Register RA) is programmed as an output in line 13.

Next comes a loop in which port pin RA.0 is set and cleared again. As you probably know, this port line controls LED 1. A subroutine named 'time' is called in between the LED on and LED off instructions. It consists of three nested loops that provide the necessary delay.

NEXT MONTH

In next month's fourth and penultimate instalment of the series we will discuss further programs and applications for the SX microcontroller.

(990018-3)